**1. Introduction**

1.1. **Purpose**

The purpose of this document is to detail the requirements for the Avi system to be developed. It will describe the features, functions and interfaces of the system. This document is intended for developers of the system as this document will serve as a guideline for the development of the Avi system. It is also intended for the client(The University of Witwatersrand) and serves as an indication of what the client should expect from the developed system. The approval of this document will be interpreted as permission to go ahead with the development of the system.

1.2. **Scope**

Avi is a web application that will recommend elective courses to prospective postgraduate students and further predict the grades the students are likely to get in the recommended courses based on their grades in previously completed courses.

A student will start by creating an account which includes specifying their firstname, lastname, student number and their login password.

Once a student has created an account they can only be able to get recommendations once they have added the courses they have done, the year the course was completed along with the grade they obtained in each course. This information can be updated by the student for cases where they may have made a mistake or their mark has changed.

The student can now request to get recommendations. After requesting to get a recommendation, the student must specify the level of study they want recommendations for. After this the system will present to them the recommended courses along with the predicted grade for each course.

1.3. **Definitions, acronyms and abbreviations**

1.4. **References**

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998

1.5. **Overview**

The next section, section 2, of this document gives an overview of the system’s functionality. It does not specify the requirements but rather describes the factors that affected the choice of requirements and how the system operates inside various constraints. Section 3 describes all the software requirements. It comprises of detailed and technical descriptions of the system’s functionality (i.e. inputs, outputs).

**2. Overall Description**

2.1 **Product Perspective**

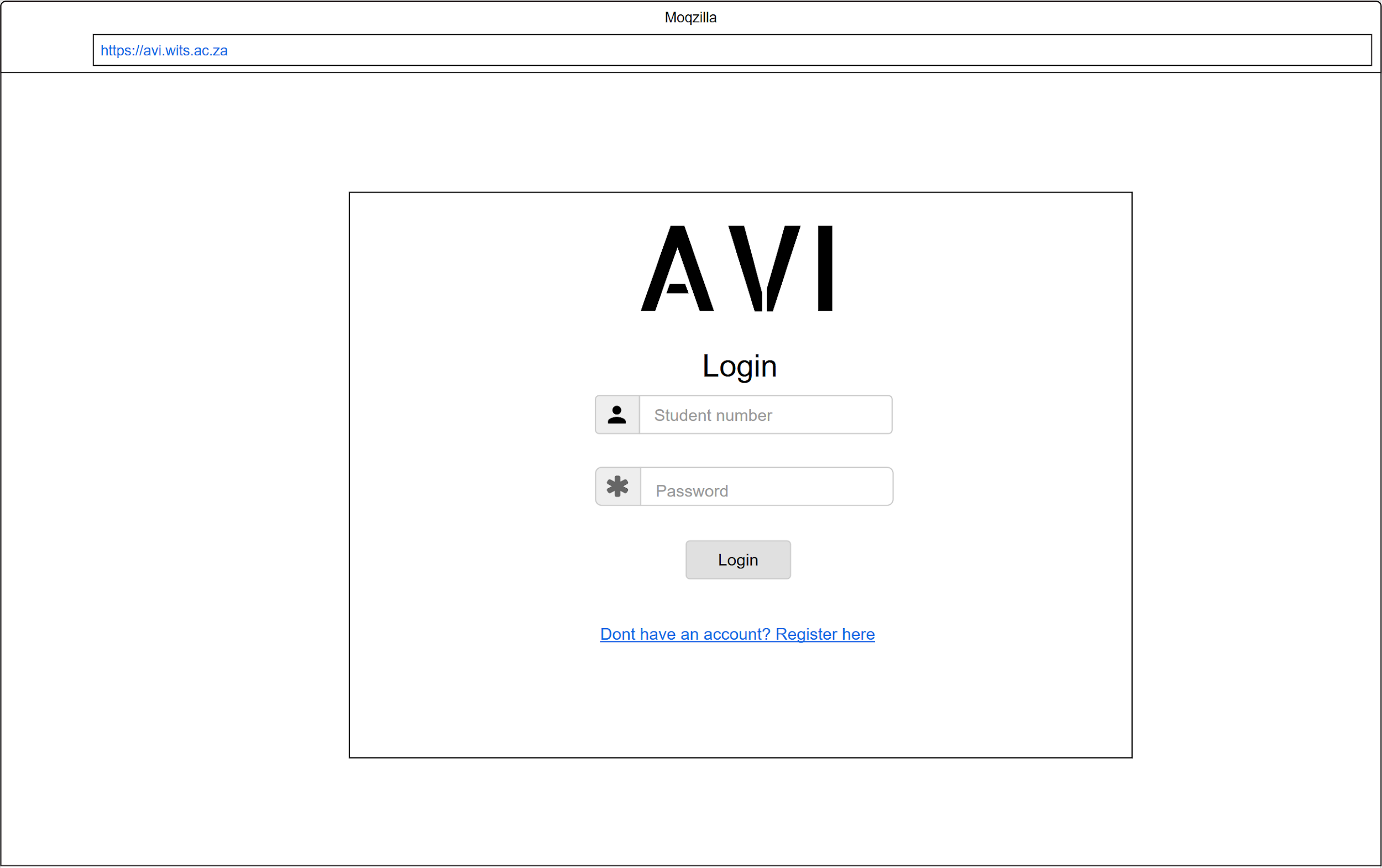
The Avi system is the first of its kind at the University of Witwatersrand. It is independent and totally self contained. Currently recommendations are provided by lectures based on their observations of other students’ performance there is no formal or reliable system or methods for providing these recommendations.

\*\*\*Use Case Diagram goes here\*\*\*

2.1.1. System Interfaces

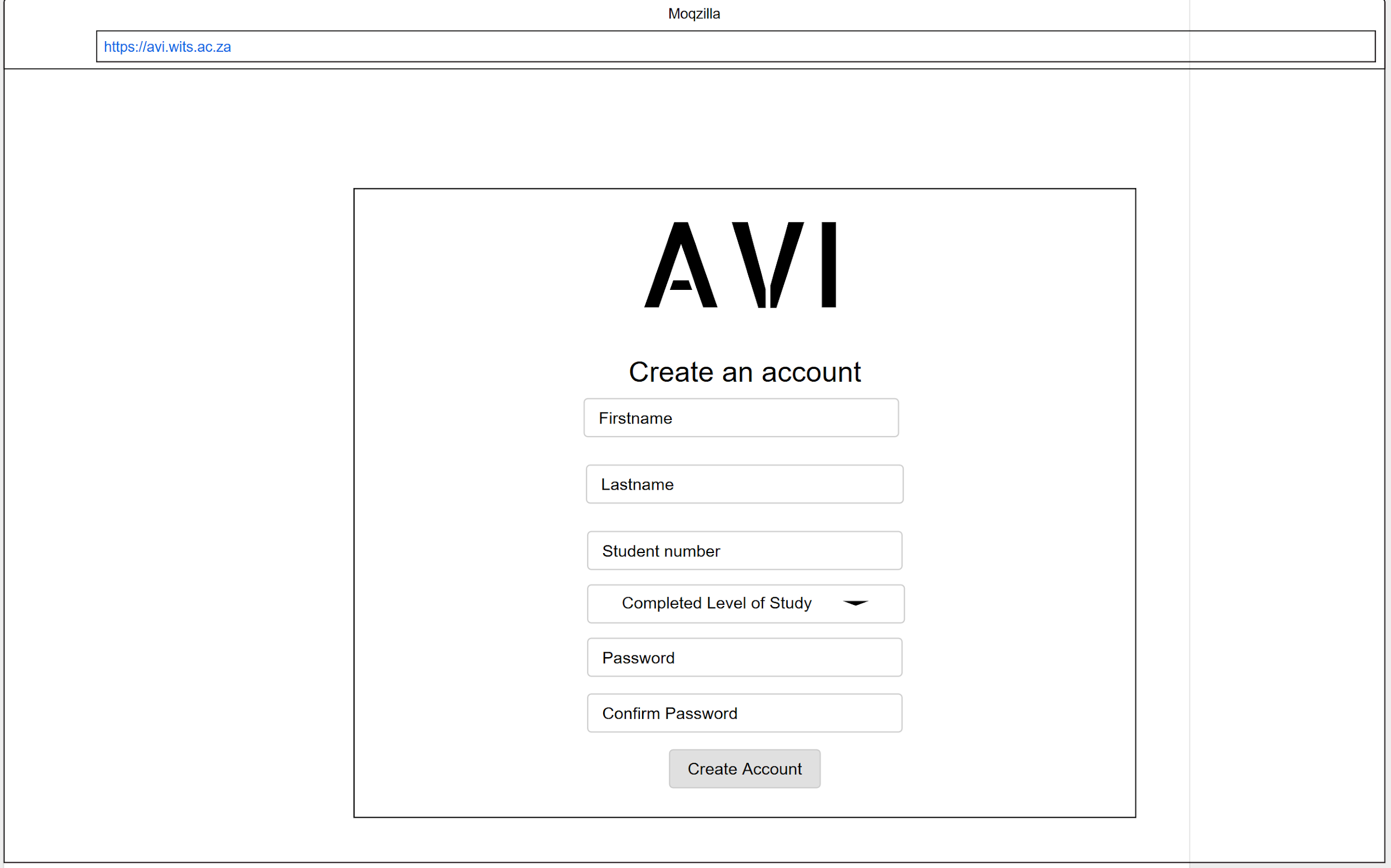
2.1.2. User Interfaces

Login Page:



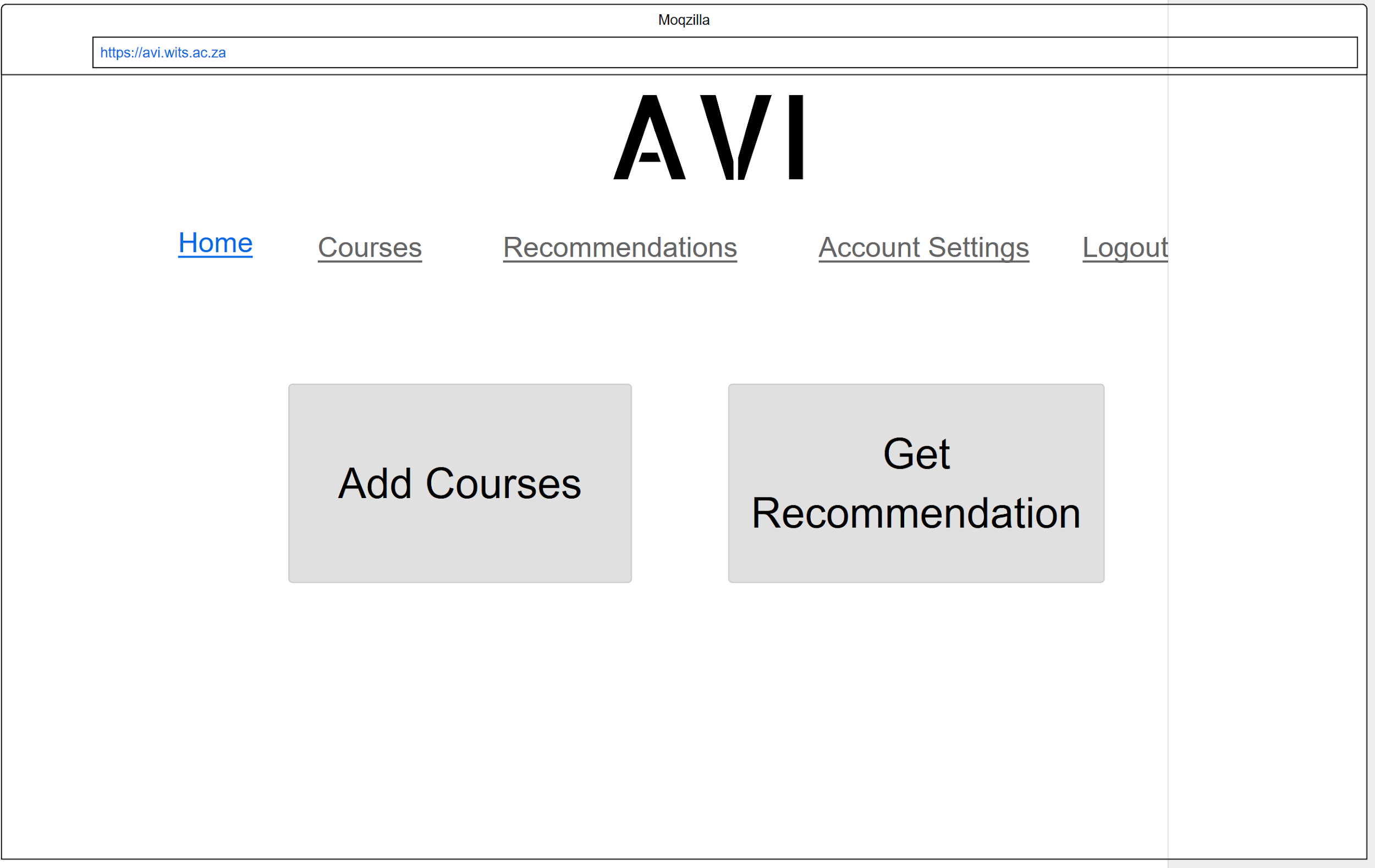
This is the first page that any user sees. If the user already has an account they should be able to login by entering their student number and password in the provided fields and clicking on the ”Login” button which would then direct them to the Home Page. A user that has not created an account should be able to click on the “Don’t have an account?Register here” link which will direct them to the Create Account Page.

Create Account Page:



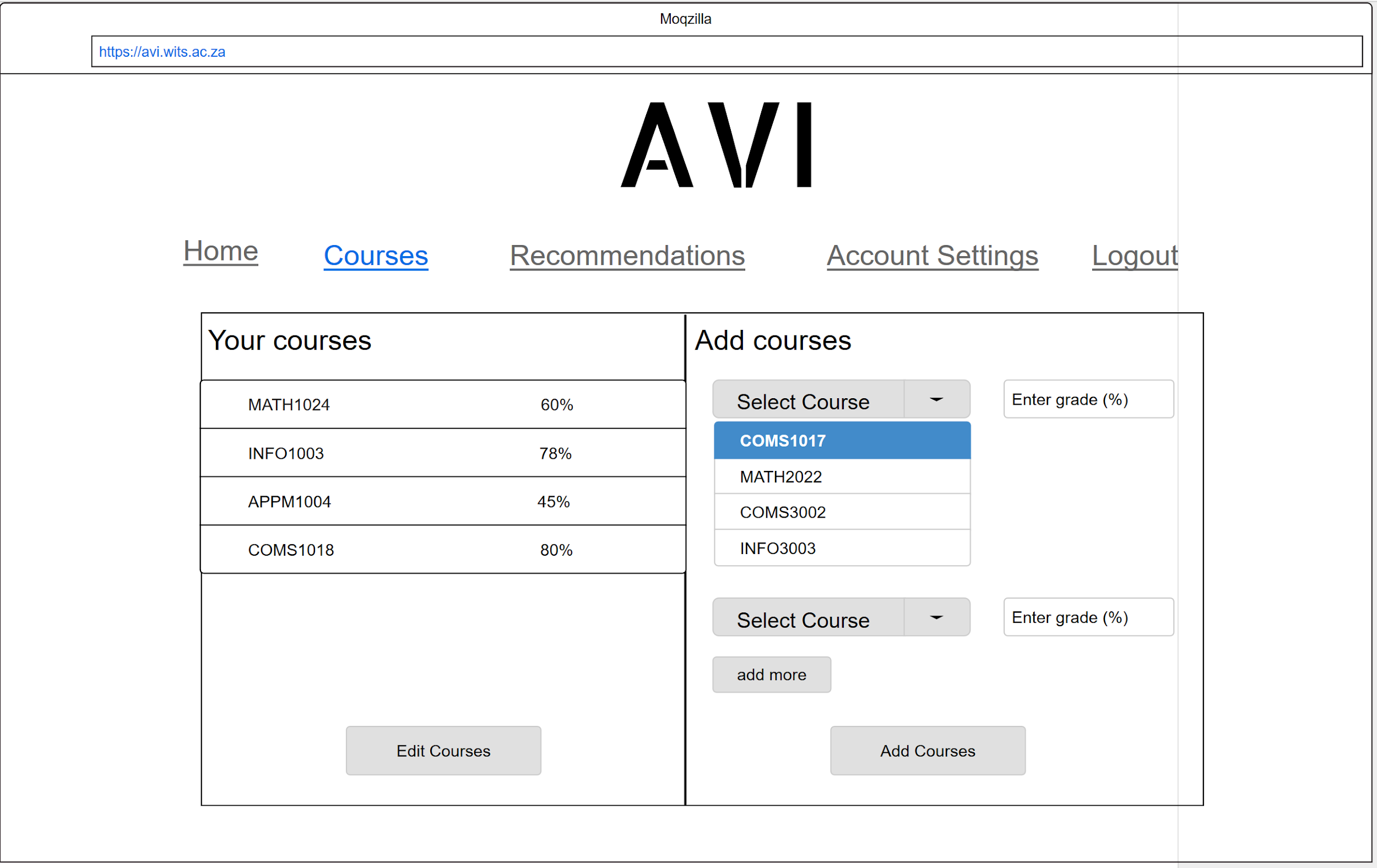
On this page the user should be able to create an account by filling in their Firstname, Lastname, Student number, Completed Level of Study, setting their password and then clicking on the “Create Account” button. Once they create account button has been clicked, the user will be taken back to the Login Page so that they can login.

Home Page:



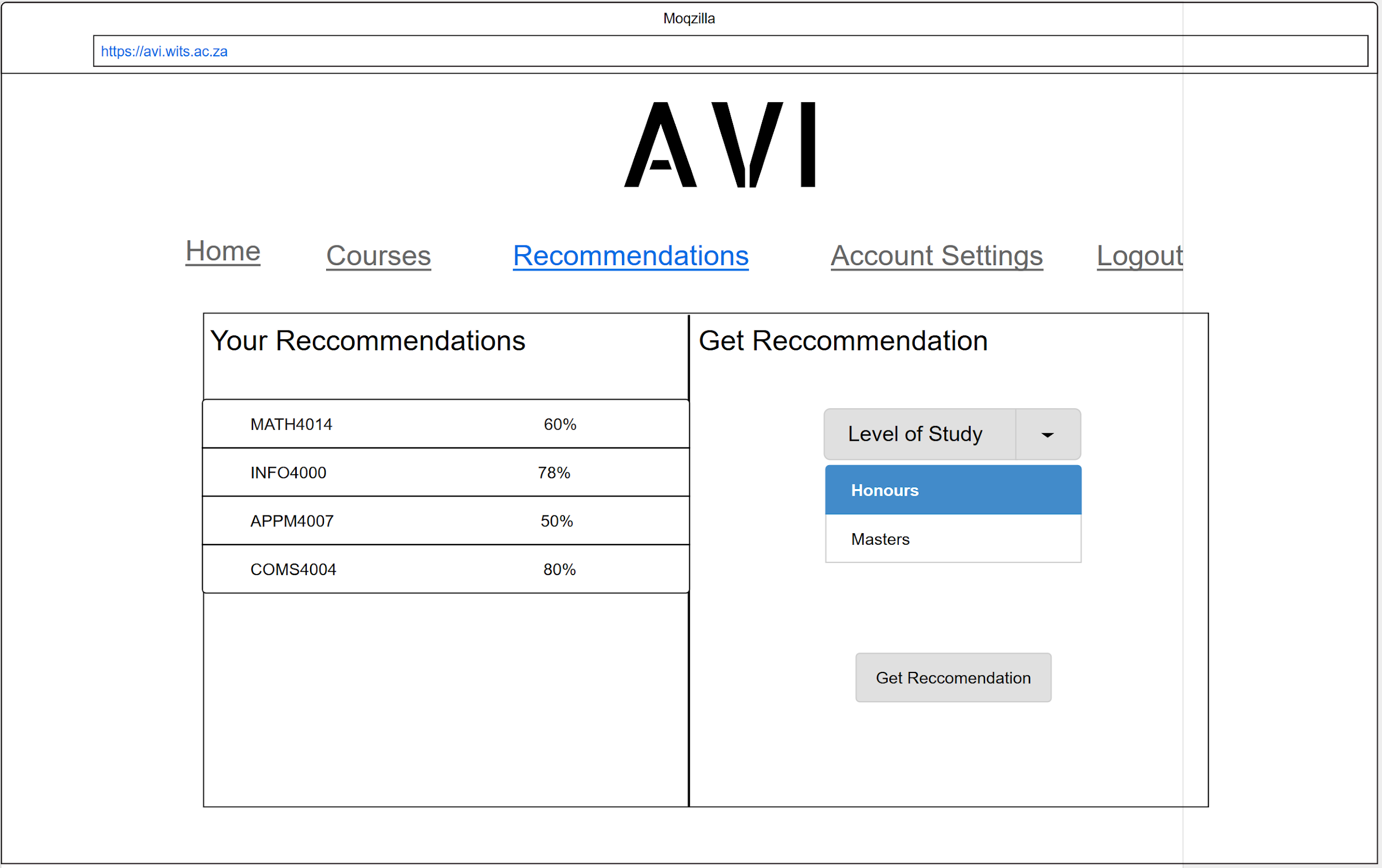
Once the user has logged in, this is the page they will see. On this page they can either choose to view their courses by clicking on “Courses” in the menu, view their recommendations by clicking the “Recommendations” in the menu, add courses by clicking on the “Add Courses” button or get recommendations by clicking the “Get Recommendation” button. When the “Add Courses” button or “Courses” is clicked, the user will be directed to the Courses Page and when the “Get Recommendation” button or “Recommendations” is clicked, the user will be directed to the Recommendations Page. Note that only users who have added courses will be able to get recommendations.

Courses Page:



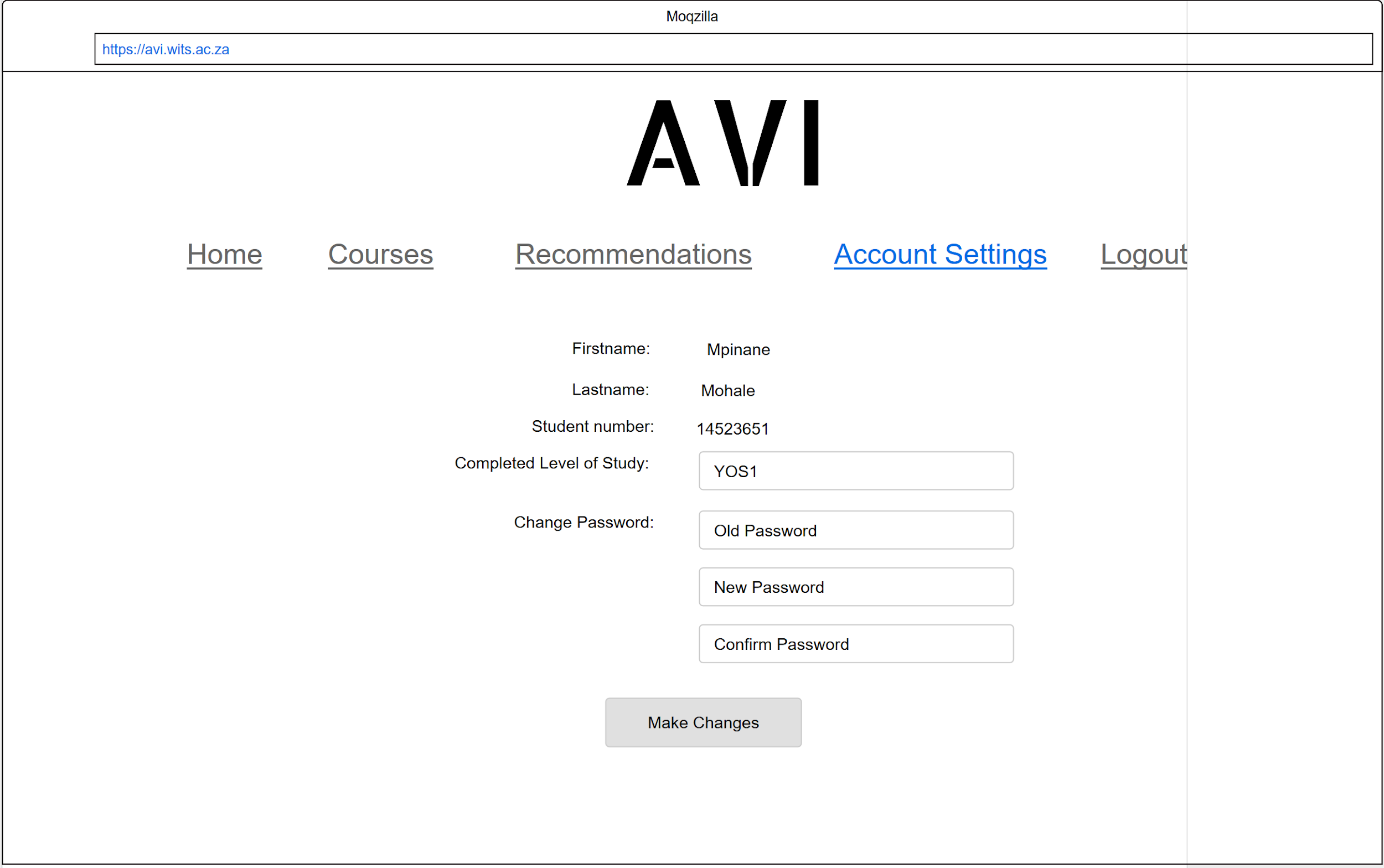
On this page the user can see the courses they have entered and are able to edit them. They can also add more courses on the right side of the page by entering the course coode and the grade obtained in the course. Multiple courses can be added at once by clicking the “add more” button so that more input fields are generated. Once the user has entered the courses and their grades, they must click on the “Add Courses” button and the courses will be successfully added and will appear under “Your courses”.

Recommendations Page:



On this page the user can see their recommended courses and they can request to get a recommendation. To get a new recommendation the user must specify a level of study and then click on the “Get Recommendation” button. Once the button has been clicked, the recommended courses will appear under “Your Recommendations”

Account Settings Page:



This page is reached by clicking on “Account Settings” in the menu. On this page the user can change their completed year of study and/or their password. To change their completed year of study they must enter a new value in the provided field and then click the “Make Changes” button. To change their password they must provide their old password as well as the new password in the provided fields and then click the “Make Changes” button. To change both the year of study on password at the same time, the user must perform the above 2 steps and only click the “Make Changes” button once all necessary fields have been filled in.

2.1.3. Hardware Interfaces

Client Side:

The user needs a device that has internet connection and one of the browsers mentioned in the next section.

Server Side:

Since this system will not be deployed, the hardware requirements are a laptop or desktop where the system and the web server will be stored locally.

For the database, the University of Witwatersrand’s LAMP server will be sufficient

2.1.4. Software Interfaces

Client Side:

The “Avi” system can run on any browser but the preferred browsers are Firefox, Microsoft Edge and Chrome since the system will be tested on these browsers. Any user that has a device with these browsers should be able to use the system.

Server Side:

The “Avi” system will be run on the Django Framework’s server which comes with the installation of the framework. The database for data storage and retrieval will be stored on the University of Witwatersrand MySQL server LAMP.

2.1.5. Communication Interfaces

The “Avi” system uses http/https protocol for communication (over the Internet) with the MySQL database and TCP/IP for intranet communication with the backend hosted locally on a laptop or desktop.

2.1.6. Memory Constraints

Client Side:

None

Server Side:

The computer that will be storing the system locally will need enough storage to store the system. Currently the size of the system can not be determined since it has not yet been developed.

\*\*\*

THIS SEEMS OUT OF PLACE

1.Product Perspective

The software is independent and self-contained.

User access is permitted on any system with a functional operating system which has a browser, this automatically makes most recent hardware system included.

The site does not require any specific adaptation either than those specified and required by the browser.

The logistic characteristics of the interface follow those specified by each browser, which may or may not be unique. (screen format, pages, window layout etc.)

Software interface such as importing mathematical libraries will be required for the calculation-based aspect of the software – machine learning implementation.

The software will not require any direct link to other system software or hardware components.

The network protocol used by the software will be http/ https.

Primary and Secondary memory limits will be handled by the kernel during browser runtime and will not be in any way directly invoked by the software.

User initiated operations will be permitted such as database manipulation through account creation, login attempts and more. User credentials will be stored with an option to recover and reset password or delete account.

\*\*\*

2.2. **Product Functions**

? ?

2.3. **User Characteristics**

The user will be expected to have completed or have some amount of tertiary education, which would entail the ability to intelligibly use the software with ease.

2.4. **Constraints**

The software will impose minimal constraints on its own. Any constraint will be largely attributed to the browser used instead of software limitations.